

Let's Get Cozy with Relations

by Anthony Carrico <acarrico@memebeam.org>

<https://github.com/acarrico/presentations/>



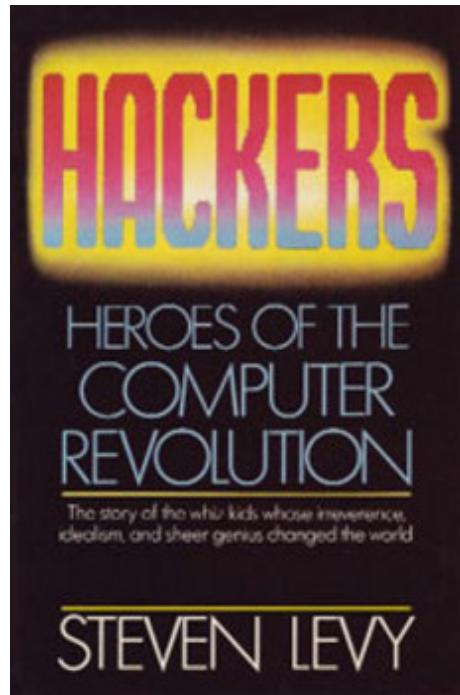
@Anthony_Carrico #vtfun @racketlang

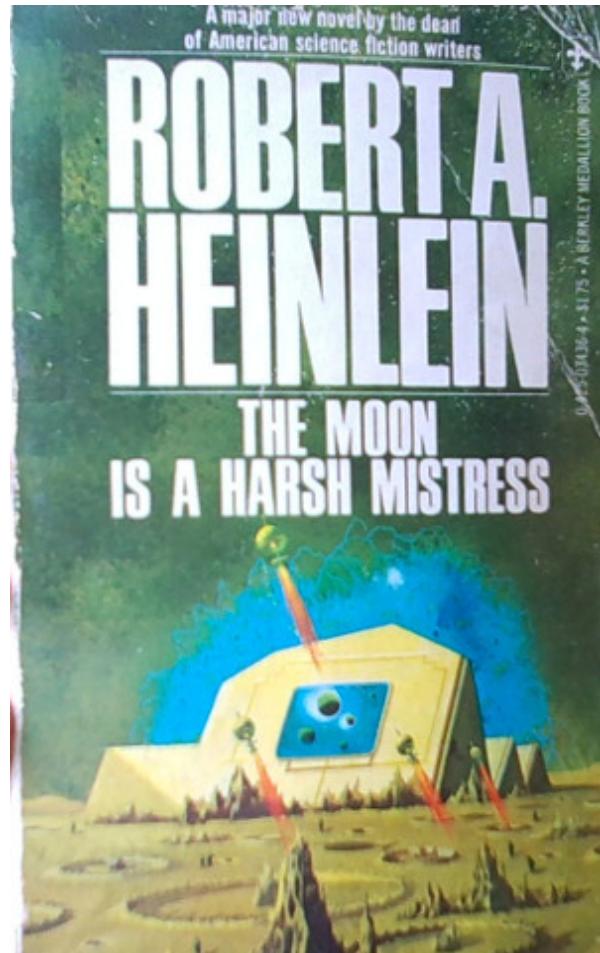
Vermont Functional Users Group

Context

- 192x λ -Calculus: Alonzo Church
 - 1937: Church–Rosser theorem
 - Brian Waters, Lambda Calculus for Devs
- 1934, 1958, 1969: Curry–Howard correspondence
- 1936: Church–Turing Entscheidungsproblem
 - Eric Smith, The Sep of Church and State
- 1958: Expressions, John McCarthy, Algol60, Lisp
 - Anthony Carrico, Pay Attntn to Racket
- 1964: Peter Landin-ISWIM, Christopher Strachey, Gordon Plotkin, ...
 - Eric Smith, The Next 700 Prog Langs

Wizards







1

in the Earthsea Trilogy
by the winner of the
Hugo and Nebula awards

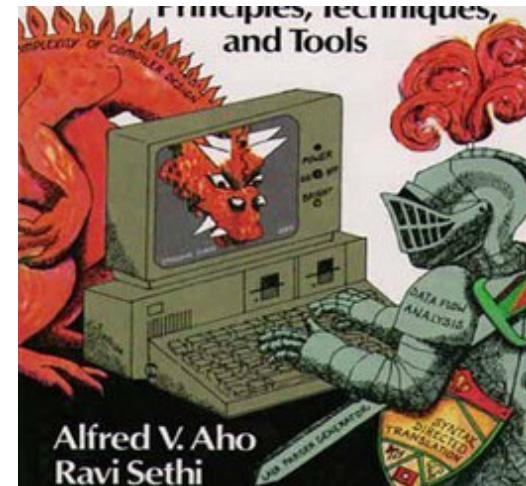
Ursula K. Le Guin
A Wizard
of Earthsea



15767-0 * \$2.25 * A BANTAM BOOK

Structure and Interpretation of Computer Programs

“It’s said that computer science is a lot like magic. And it’s sort of good that it’s like magic. There’s a bad part of computer science that’s a lot like religion.” [Abelson \(Sussman?\)](#), SICP Lecture 2b





Julie Lerman @julielerman · 2h

❤️ "@TheMichaelMoran: Just a 140-year-old tortoise wearing her new son as a hat. NBD. "



10



13

...

[View photo](#)

First Cause

“If everything must have a cause, then God must have a cause. If there can be anything without a cause, it may just as well be the world as God...”

1927: Bertrand Russell, *Why I Am Not A Christian*

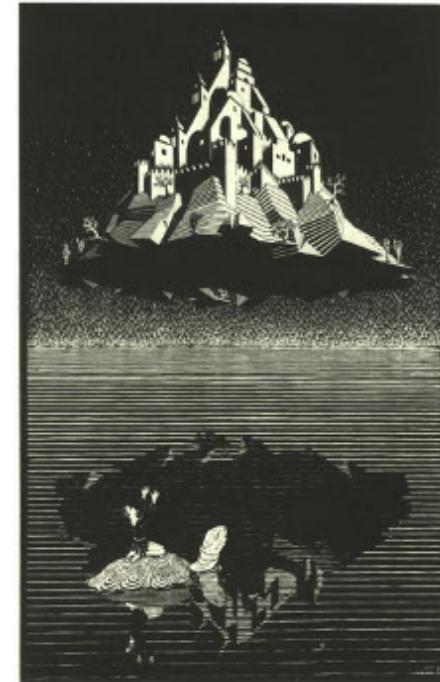
“It is exactly of the same nature as the Hindu’s view, that the world rested upon an elephant and the elephant rested upon a tortoise; and when they said, ‘How about the tortoise?’ the Indian said, ‘Suppose we change the subject.’ ”

You're very clever, but it's turtles all the way down!

Douglas Hofstadter, *Gödel, Escher, Bach*

Stephen Hawking, *A Brief History of Time*

Lev Grossman, *The Magicians*

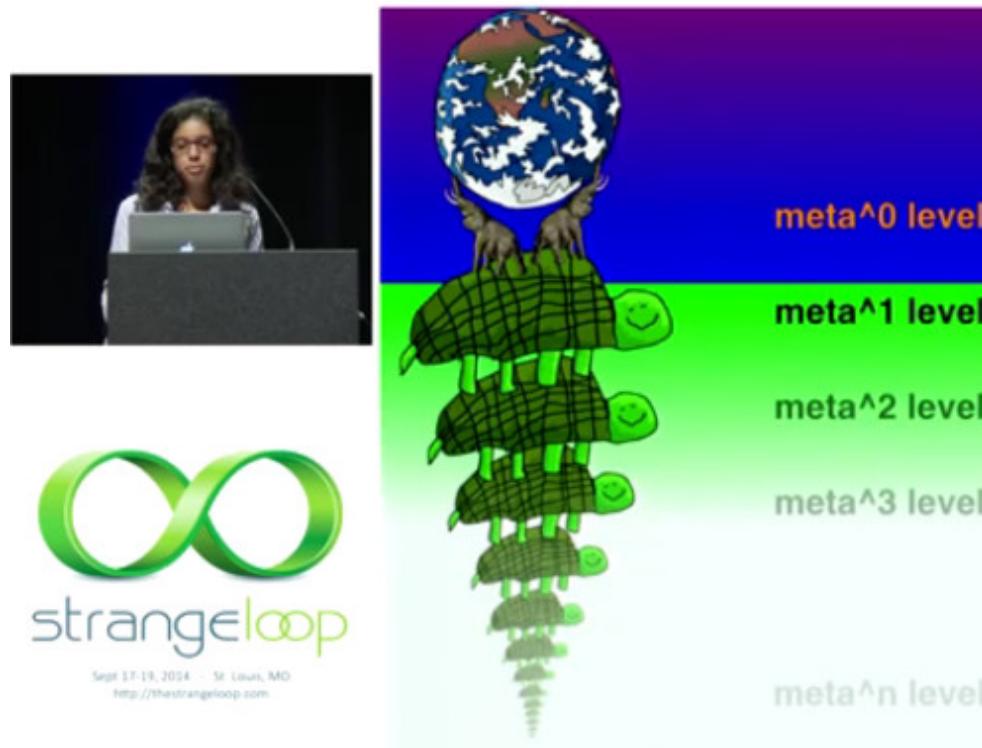


“The study of magic is not a science, it is not an art, and it is not a religion. Magic is a craft.”



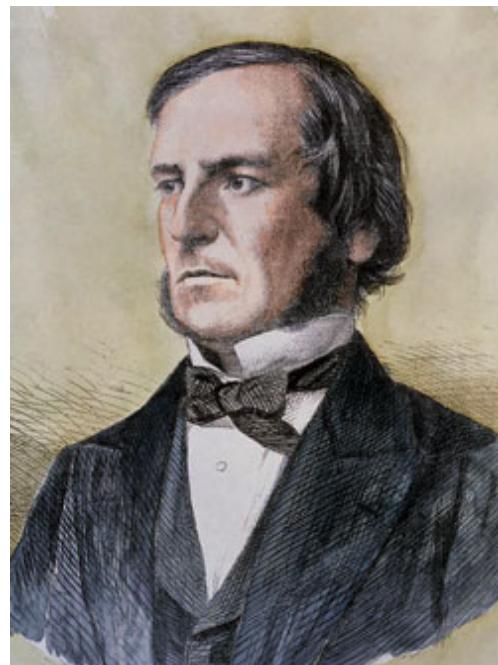
Programming Should Eat Itself

Nada Amin, Strange Loop 2014,
Programming to understand programming.

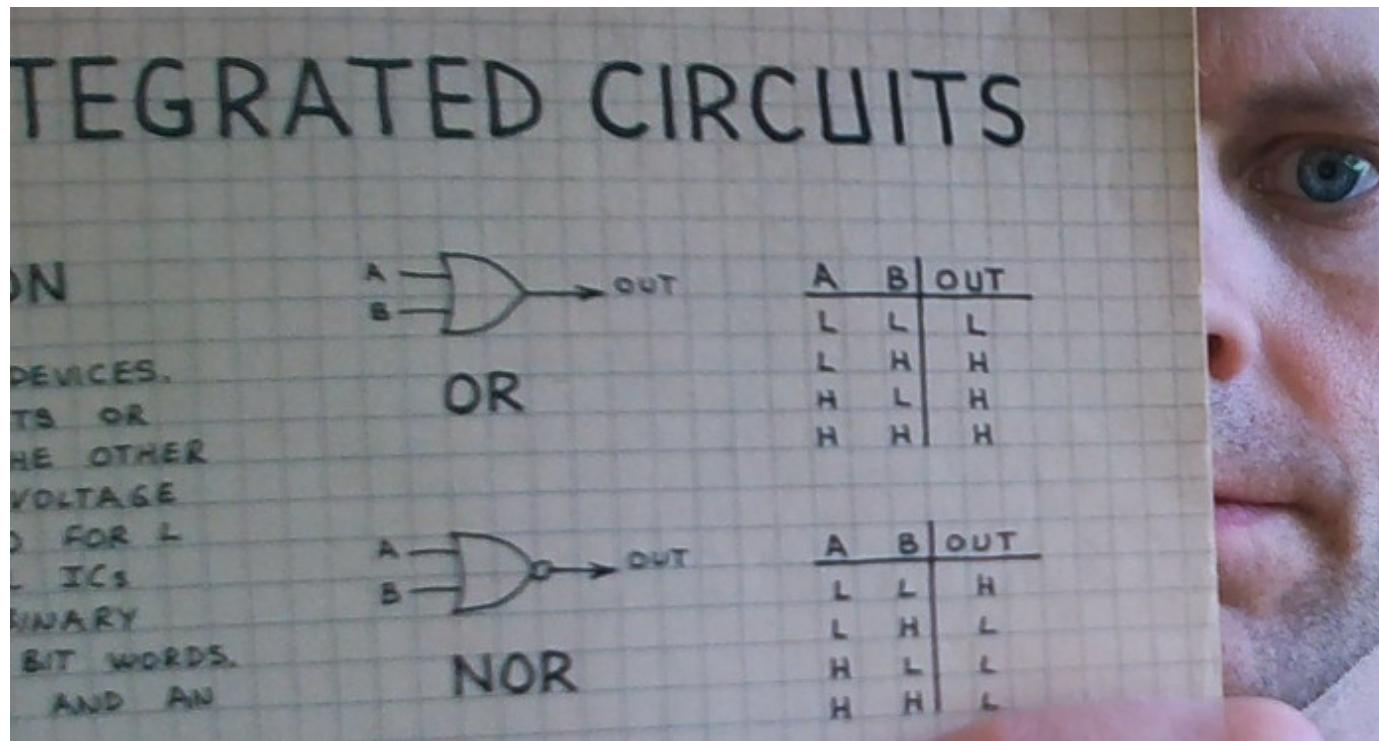


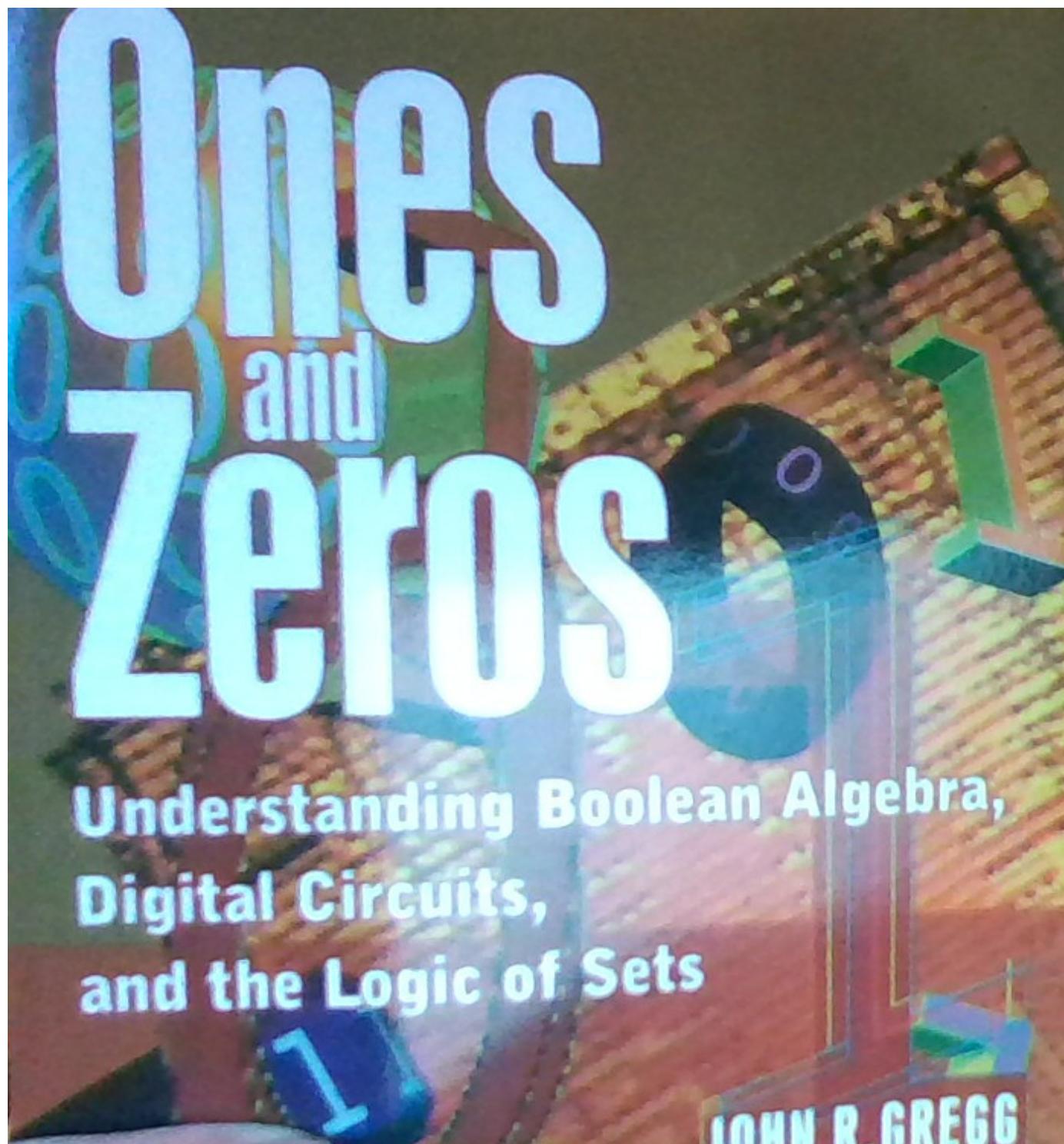
Kenichi Asai's reflective programming language
Black

George Boole (1815-1864) — Boolean Algebra

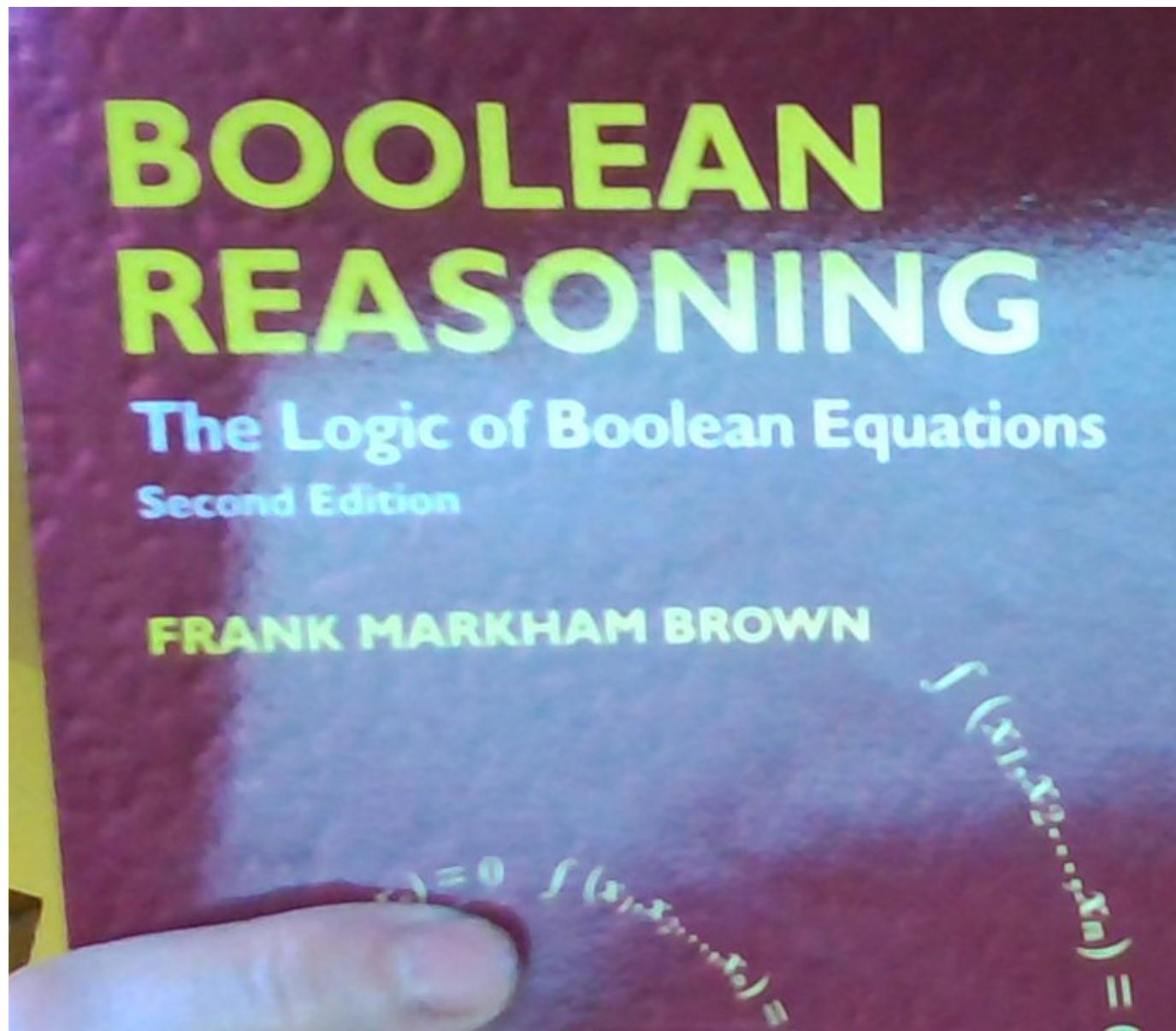


1982: Forrest Mims III, *Engineer's Notebook II*





“The operational (i.e. functional) basis of Boolean reasoning differentiates it from the predicate calculus, whose basis is relational”



Video Game Example

1998: John Gregg, *Ones and Zeros*

- No minions can teleport while wearing armor.
- Any minion worth 150 points but still vulnerable to the electro-neural disrupter can teleport.
- All 150-point minions are armored.
- Any minon that is electro-neurally disruptable and not worth 150 points is either armored or a teleporter or both.
- Any minion that is in vulnerable to both of your weapons either can teleport or is not worth 150 points or both

Good luck! The fate of humanity rests in your hands!

Blake Canonical Form

1937: Archie Blake, *Canonical expressions in Boolean algebra.*

- A minion cannot wear armor and teleport
- All electro-neurally disruptable minions are either armored or can teleport.
- No minions are worth 150 points

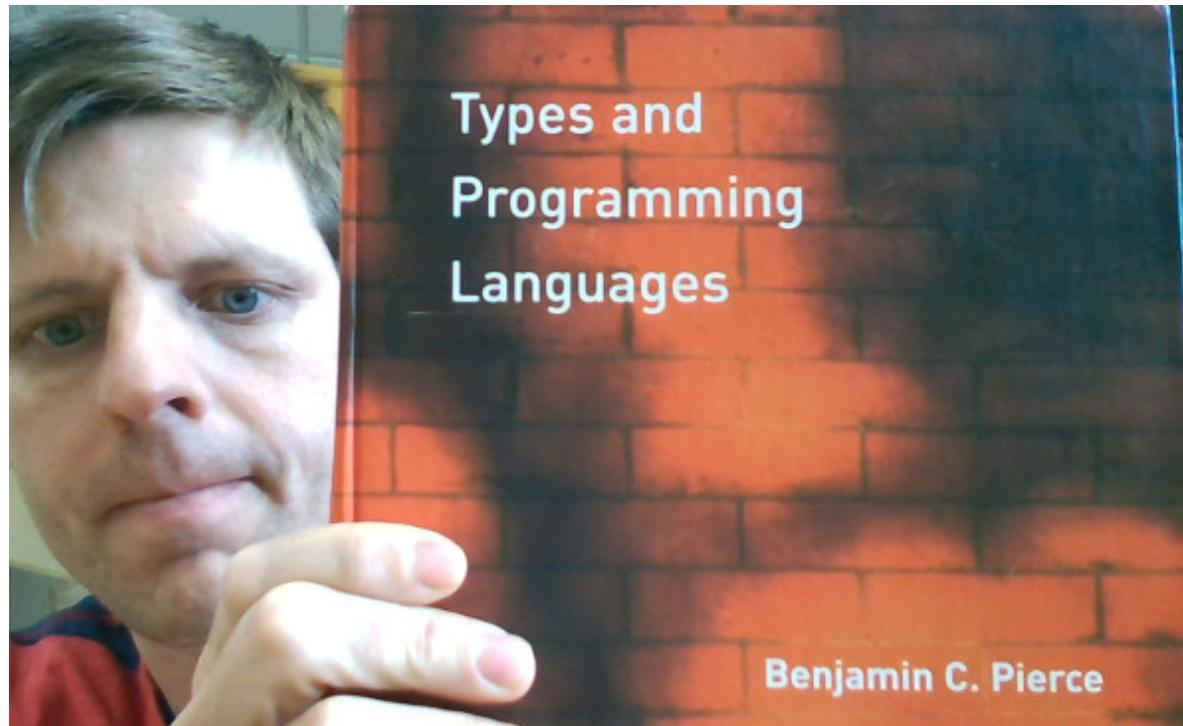
[Alan Turing](#), “The process of constructing instruction tables should be very fascinating. There need be no real danger of it ever becoming a drudge, for any processes that are quite mechanical may be turned over to the machine itself.”

Subtext 2

OOPSLA'07: Jonathan Edwards, *No Ifs, Ands, or Buts Uncovering the Simplicity of Conditionals*

Johnathan Edwards, “Schematic tables are a new representation for conditionals. Roughly a cross between decision tables and data flow graphs, they represent computation and decision-making orthogonally.”

2002: Benjamin Pierce, *Types and Programming Languages*

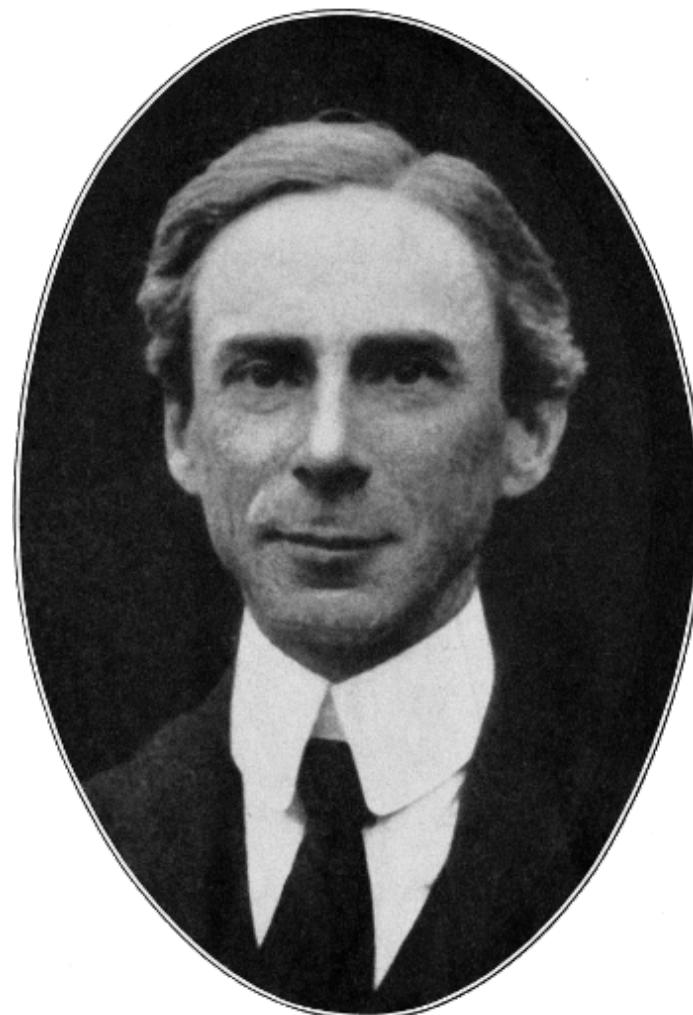


tapl relation

DEFINITION: An n -place *relation* on a collection of sets S_1, S_2, \dots, S_n is a set $R \subseteq S_1 \times S_2 \times \dots \times S_n$ of tuples of elements from S_1 through S_n . We say that the elements $s_1 \in S_1$ through $s_n \in S_n$ are *related by R* if (s_1, \dots, s_n) is an element of R . □

Bertrand Russell (1872-1970)

1910-1913: Alfred North Whitehead, Bertrand Russell, *Principia Mathematica*



Gottlob Frege (1848-1925)

- foundation of math in logic
- axiomatic, formal language
- propositional logic
- predicate logic, quantifiers



Declarative Sentence

“ *Is it true that x ?*”

1977: Wilfred Hodges, *Logic*

Proposition

2 < 10

20 < 10

“A proposition is a formula that is necessarily true or false, but cannot be both.”

1990, 2003: Frank Brown, *Boolean Reasoning*

- compare: 1 + 2
- truth-table
- propositional calculus

Predicate

$$\begin{aligned}2 &< 10 \\x + 1 &< 10 \\x + y &< 10 \\x + y &< z\end{aligned}$$

“the formula represented by $P(x_1, \dots, x_n)$ is an *n-variable predicate* if it becomes a proposition for each allowable substitution of values for x_1, \dots, x_n ”

1990, 2003: Frank Brown, *Boolean Reasoning*

- Satisfaction
- Situation: x is at least 2 years old.
- Quantification

Relation (Frank Brown)

2.6 Relations

Given two sets S and T , a *relation R from S into T* is a subset of $S \times T$. We define a predicate xRy , read “ x has the relation R to y ,” by the formula

$$xRy \iff (x, y) \in R . \quad (2.42)$$

- Brown does extend this to three or more sets

Relation

“There are two main ways to describe a relation.”

- listing the n-tuples which are in it.
- all of the ordered n-tuples which satisfy an n-place predicate

“The relation is said to be expressed by the predicate”

1977: Wilfred Hodges, *Logic*

tapl relation

DEFINITION: An n -place *relation* on a collection of sets S_1, S_2, \dots, S_n is a set $R \subseteq S_1 \times S_2 \times \dots \times S_n$ of tuples of elements from S_1 through S_n . We say that the elements $s_1 \in S_1$ through $s_n \in S_n$ are *related by* R if (s_1, \dots, s_n) is an element of R . □

- partitioning

tapl predicate

DEFINITION: A one-place relation on a set S is called a *predicate* on S . We say that P is true of an element $s \in S$ if $s \in P$. To emphasize this intuition, we often write $P(s)$ instead of $s \in P$, regarding P as a function mapping elements of S to truth values. \square

tapl binary relation

DEFINITION: A two-place relation R on sets S and T is called a *binary relation*. We often write $s R t$ instead of $(s, t) \in R$. When S and T are the same set U , we say that R is a binary relation on U . □

tapl mixfix

DEFINITION: For readability, three- or more place relations are often written using a “mixfix” concrete syntax, where the elements in the relation are separated by a sequence of symbols that jointly constitute the name of the relation. For example, for the typing relation for the simply typed lambda-calculus in Chapter 9, we write $\Gamma \vdash s : T$ to mean “the triple (Γ, s, T) is in the typing relation.” □

Shapes

- reflexive, symmetric, transitive
- equivalence
- partial-order
- one-to-one, onto, injection, bijection, surjection
- function

Multiset Discrimination

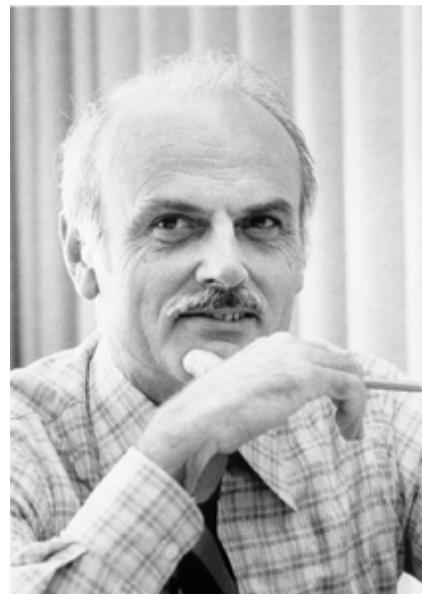
“finding duplicates in linear time without hashing or comparison-based sorting.”

2003: Fritz Henglein, *Multiset Discrimination*

- Robert Paige
- acyclic (atoms, references, structured values (ordered, sets, bag/multiset): worst case linear time
- cyclic stores: worst case time $O(m \log n)$ space $O(m)$ for m edges, n nodes
- sorting in linear time, with a sorting guide

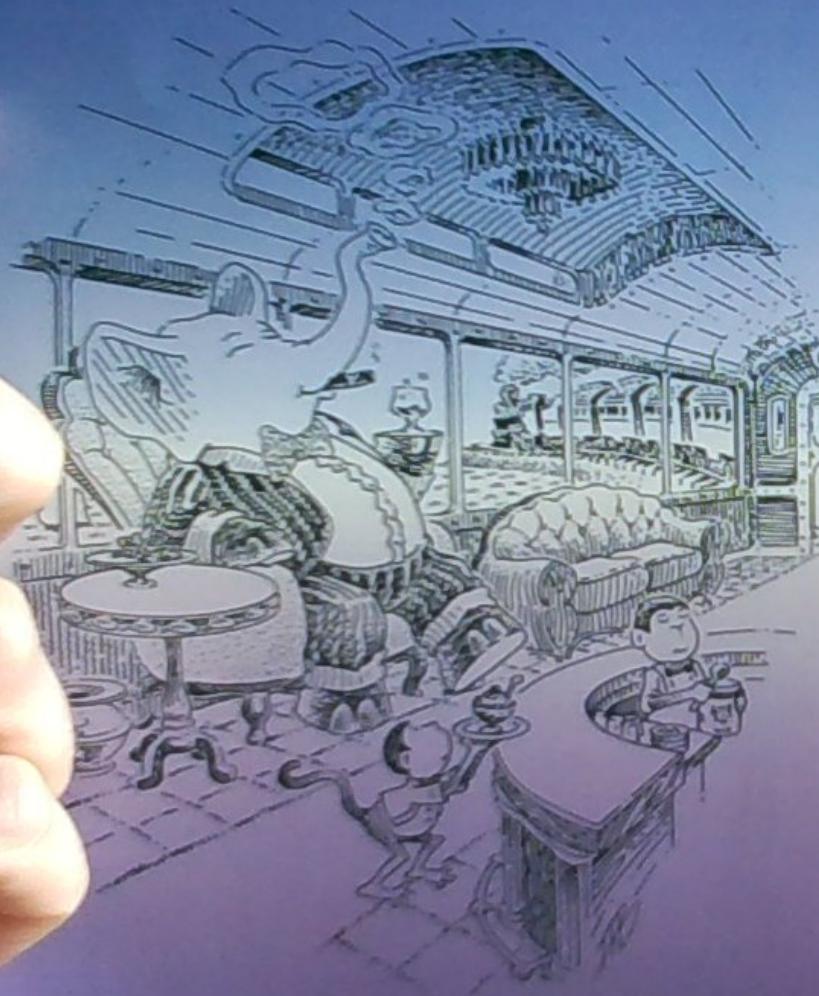
Edgar Codd (1923-2003)

1970: Edgar Codd, *A Relational Model of Data for Large Shared Data Banks*



2011: Fritz Henglein, *Generic Multiset Programming with Discrimination-based joins and symbolic Cartesian products.*

The Reasoned Schemer



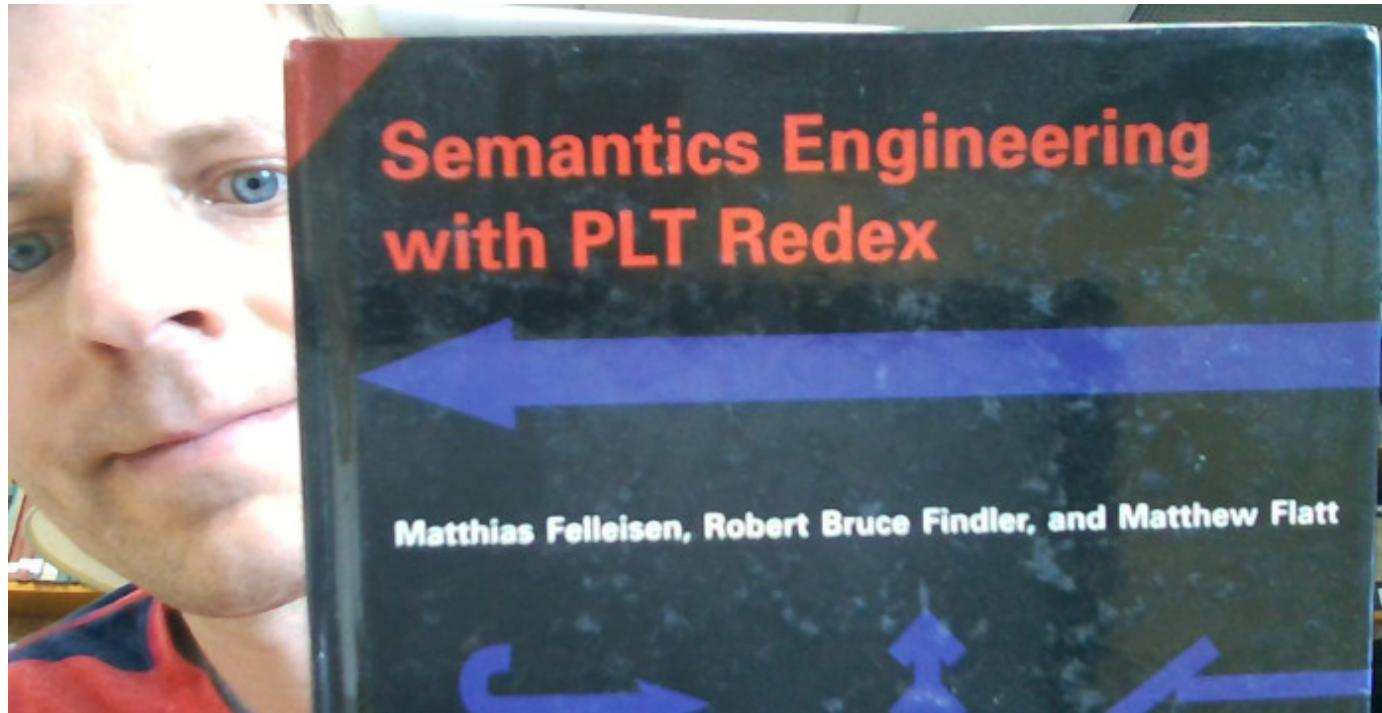
Concepts, Techniques, and Models of Computer Programming

PETER VAN ROY and SEIF HARIDI

Eastern
Economy
Edition



2009: Matthias Felleisen, Robert Findler, Matthew Flatt, *Semantics Engineering with PLT Redex*



- lambda, amb, ISWIM, CC, SCC, CK, CEK, SECD, CESK, garbage collection, typed ISWIM

Semantics via Syntax – eight pages!

name	definition	intuition
$-$	the base relation on members of an expression grammar	a single “reduction” step with no context
$\rightarrow -$	the compatible closure of $-$ with respect to the expression grammar	a single step within a context
$\rightarrow\!\!- \quad$	the reflexive-transitive closure of $\rightarrow -$	multiple evaluation steps (zero or more)
$= -$	the symmetric-transitive closure of $\rightarrow\!\!- \quad$	equates expressions that produce the same result
$eval -$	a relation projected to a range (results)	complete evaluation based on $\rightarrow\!\!- \quad$ or $= -$
$eval_-$	a generic $eval$ relation	

bool-lang

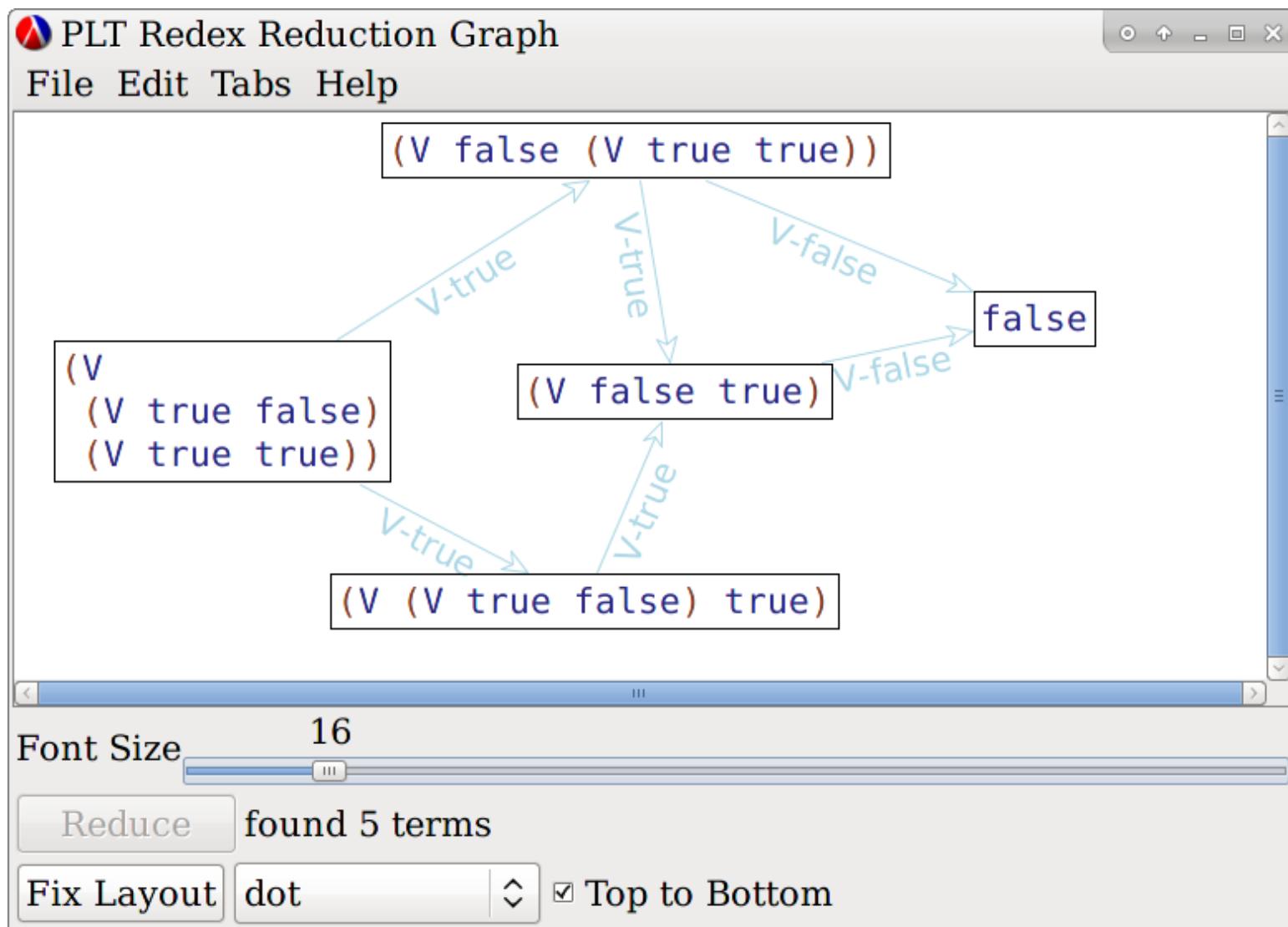
```
#lang racket
(require redex)

(define-language bool-lang
  (B true false (V B B))
  (C (V C B) (V B C) hole))

(define bool-red
  (reduction-relation
    bool-lang
    (--> (in-hole C (V false B))
          (in-hole C false)
          V-false)
    (--> (in-hole C (V true B))
          (in-hole C B)
          V-true)))

(traces bool-red
  (term
    (V (V true false)
      (V true true))))
```

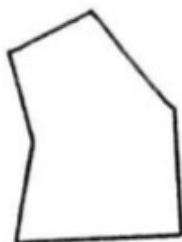
bool-lang traces



Constraints

1963: Ivan Sutherland, *Sketchpad, A Man-Machine Communication System*

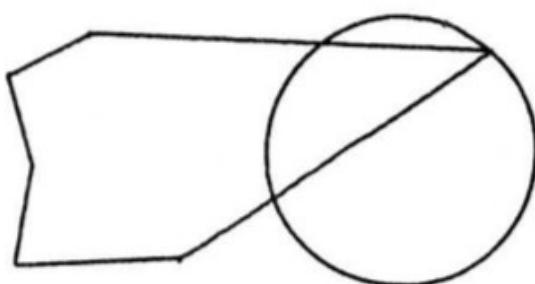
Ivan Sutherland, “Armed with the tools for representing and doing arithmetic for constraints I went gaily ahead with programming.”



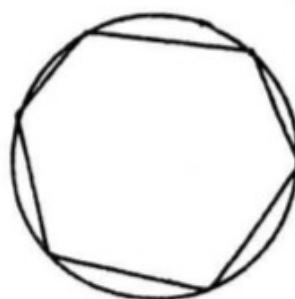
A. SIX SIDED FIGURE



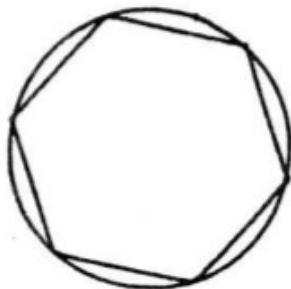
B. TO BE INSCRIBED IN CIRCLE



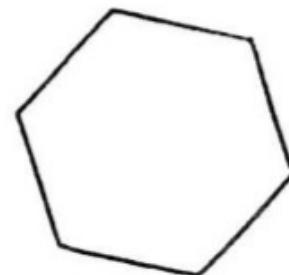
C. BY MOVING EACH CORNER



D. ON TO CIRCLE

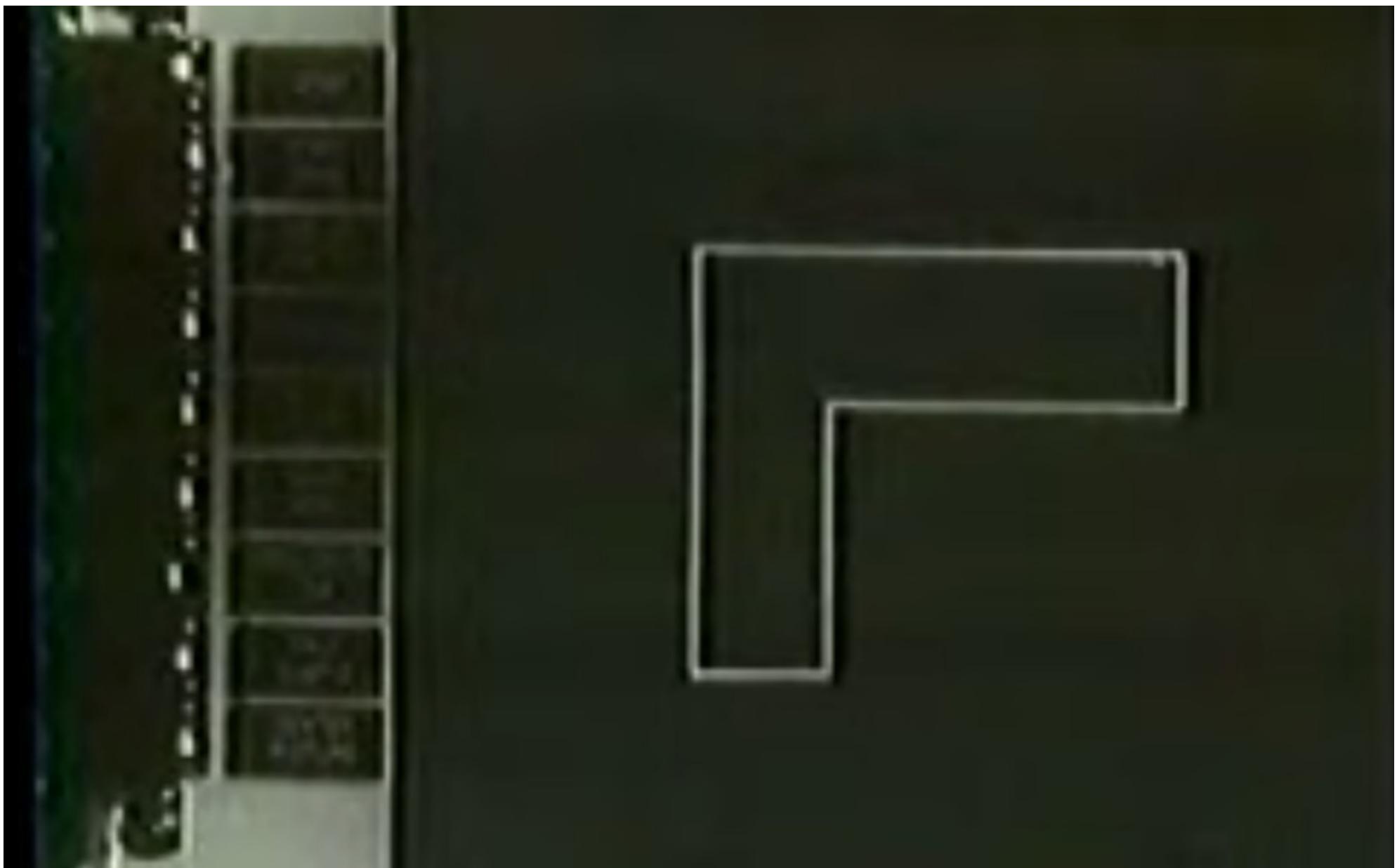


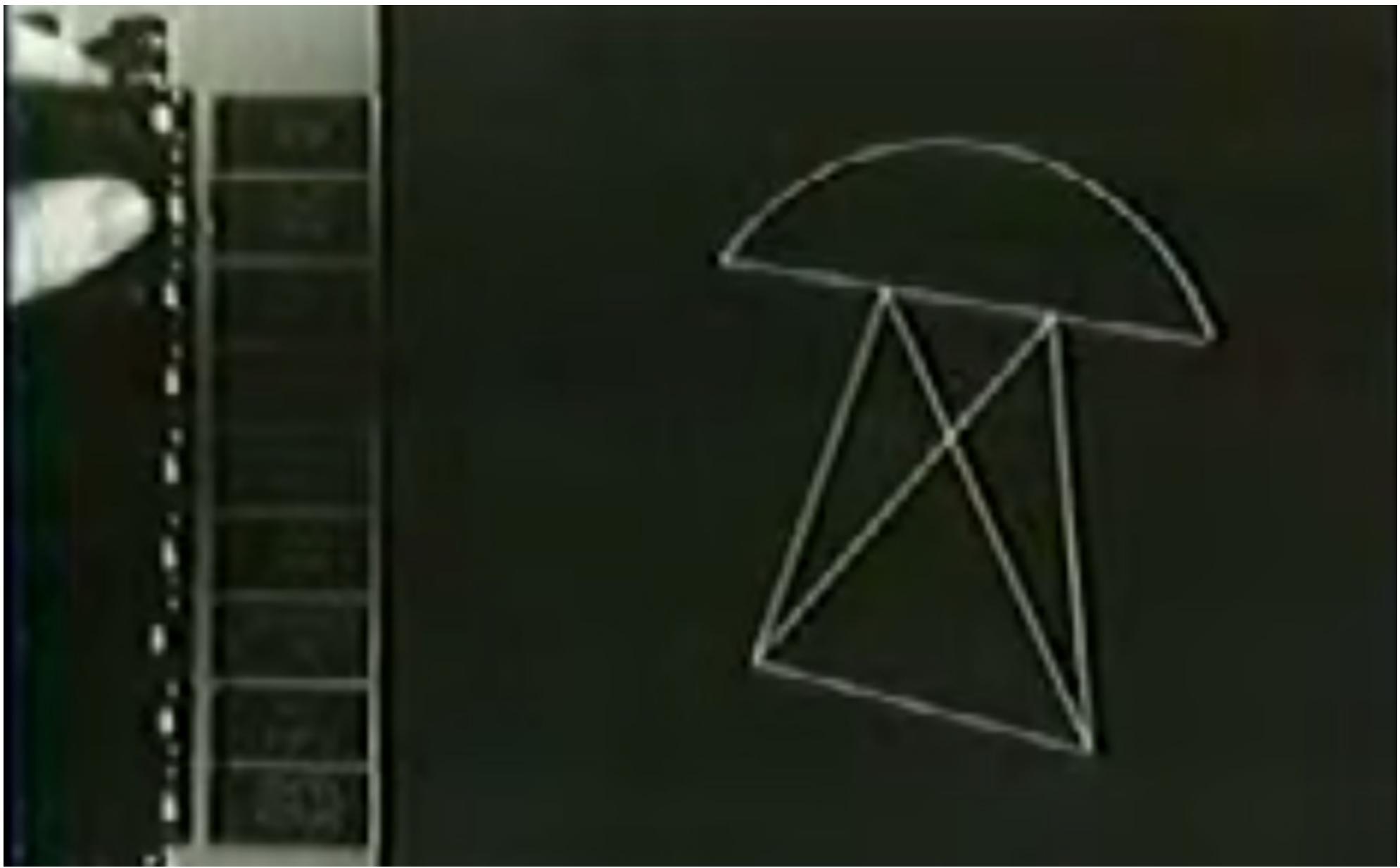
E. MAKE SIDES EQUAL

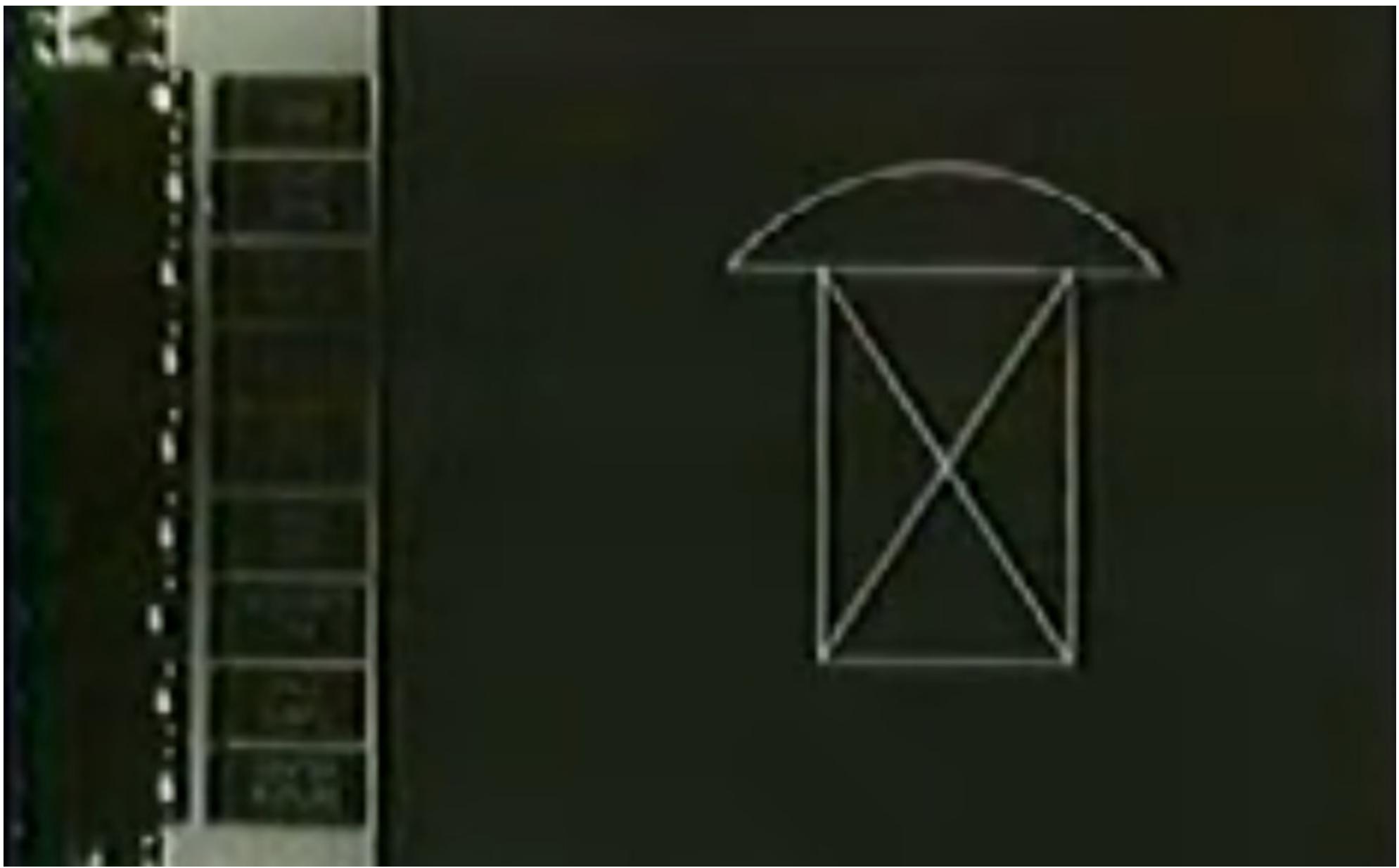


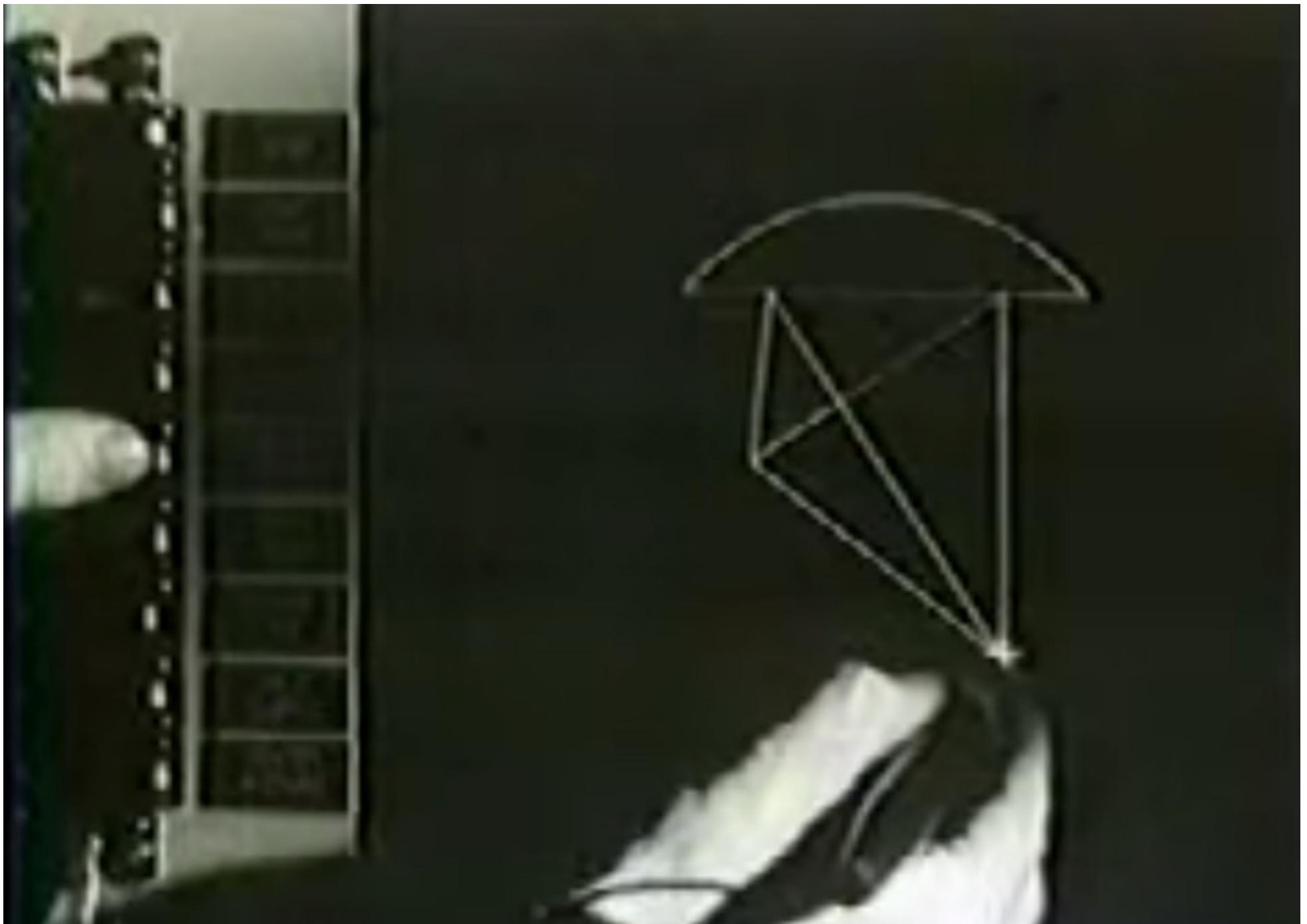
F. ERASE CIRCLE

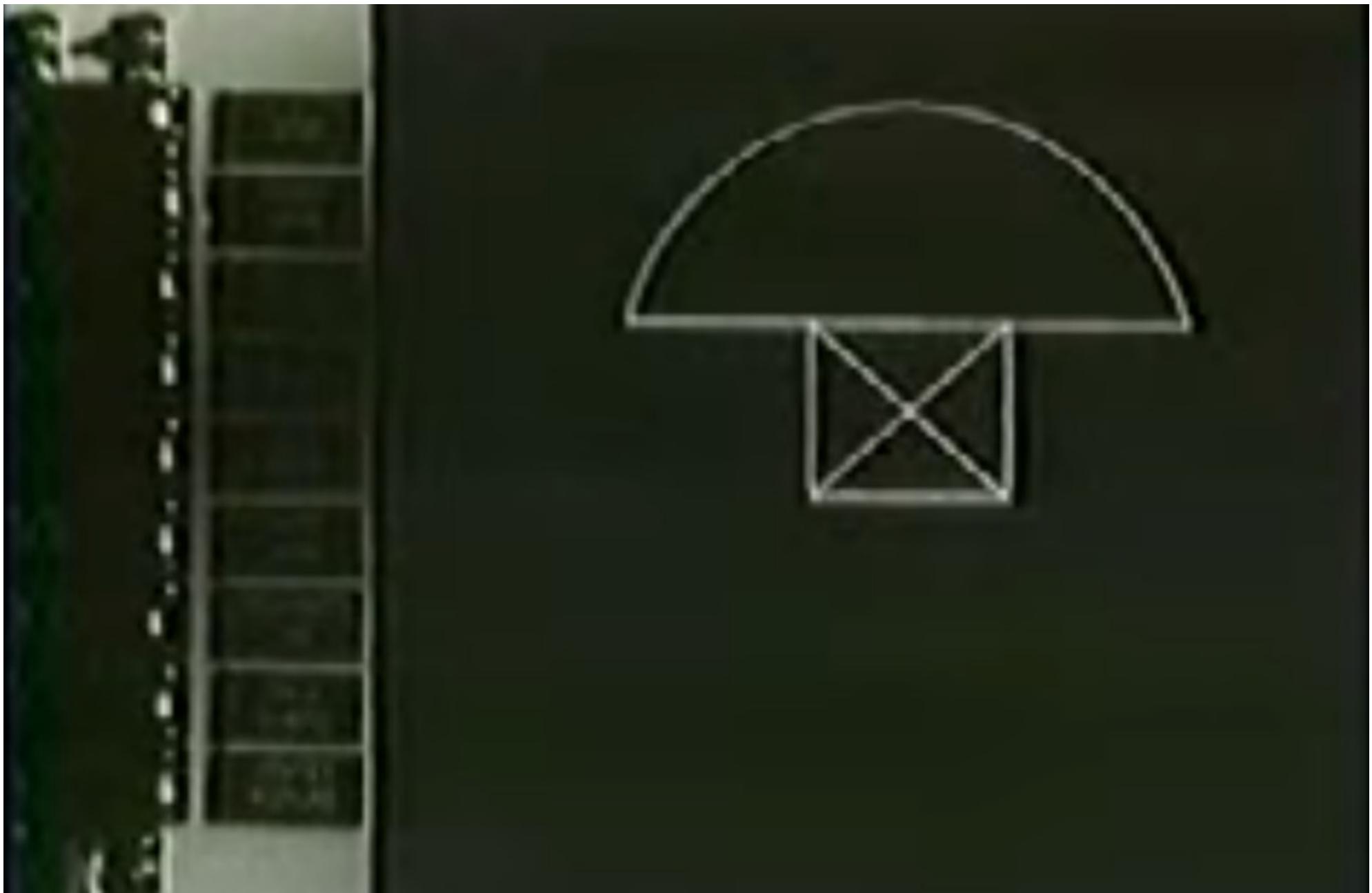


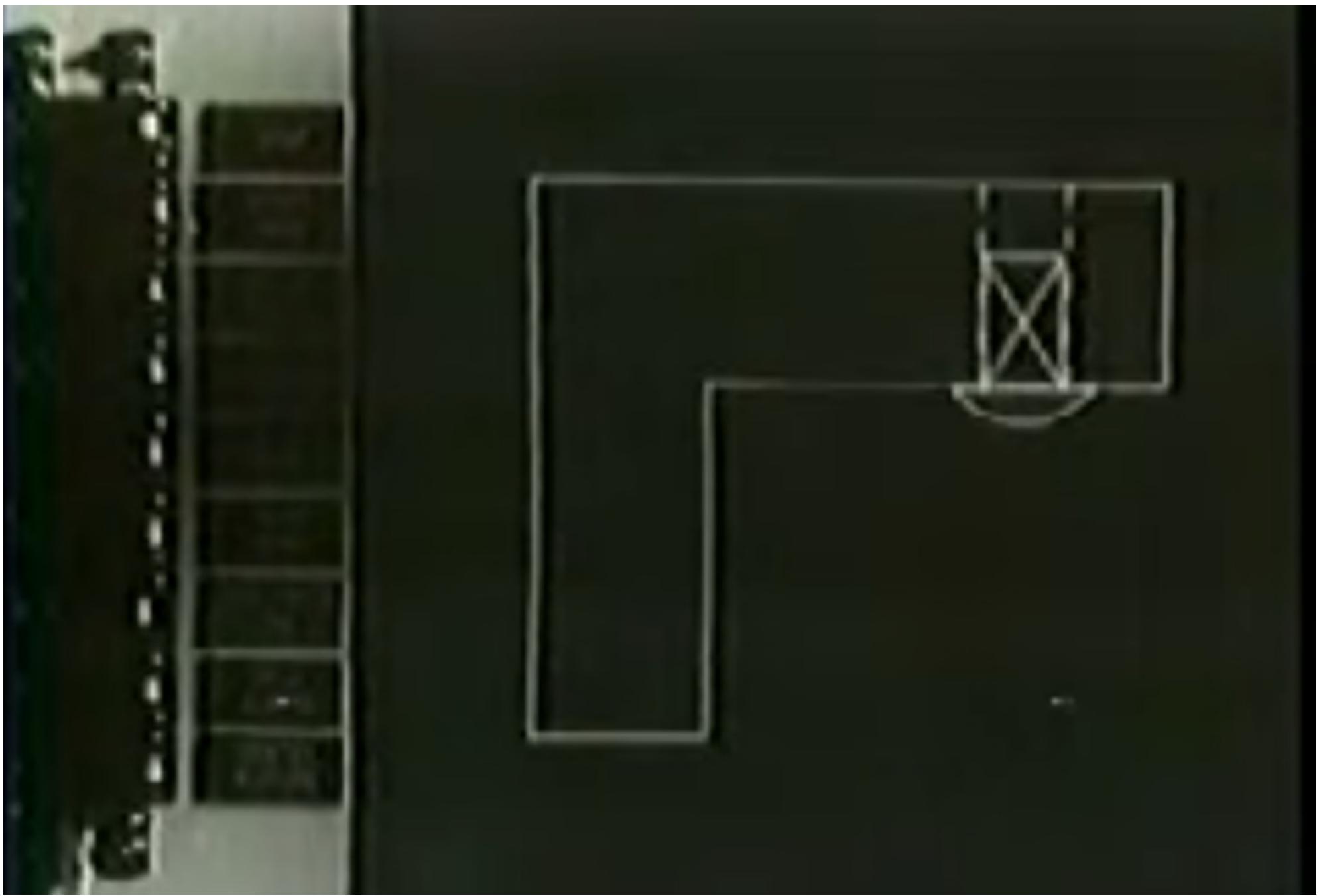


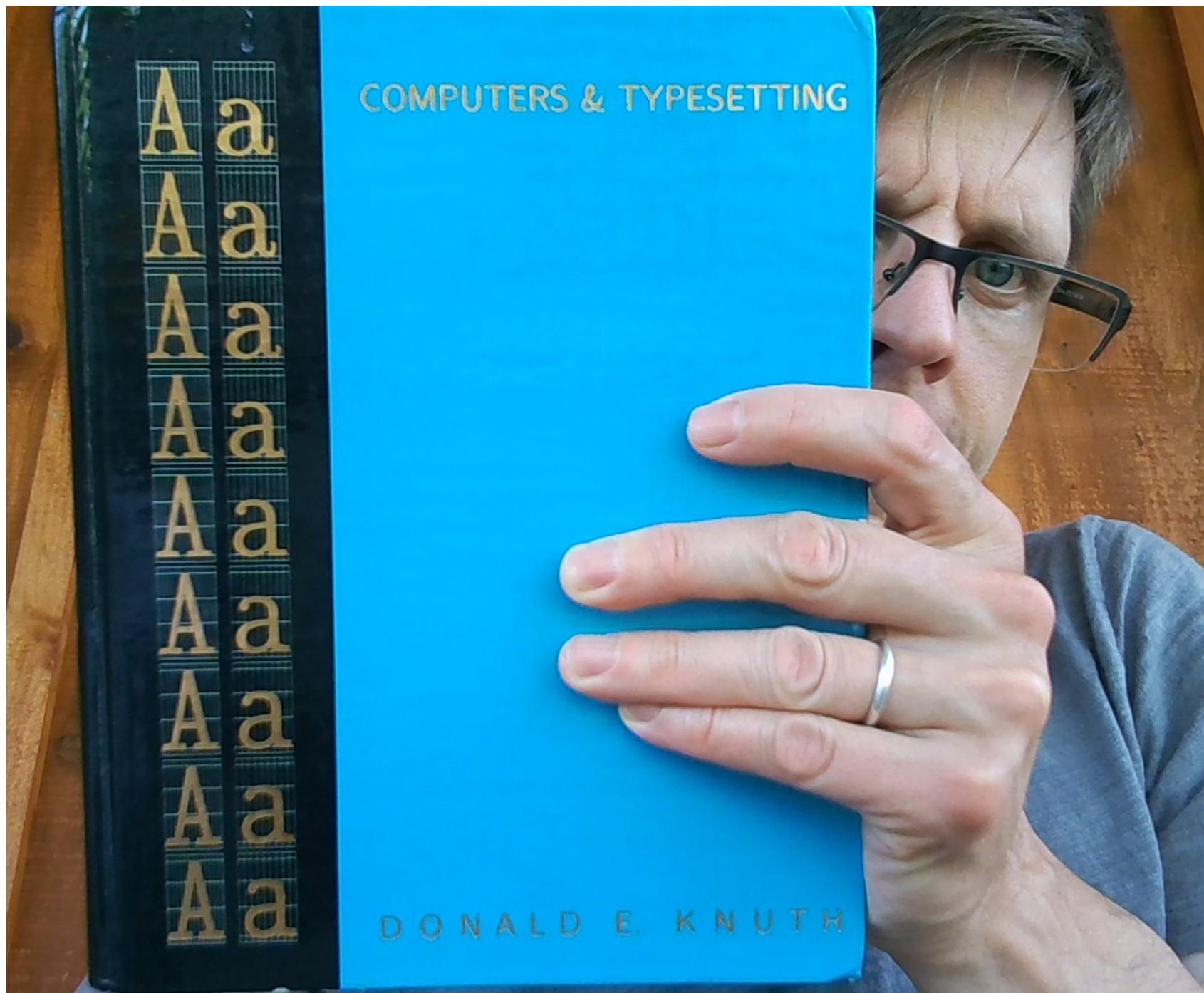












Alan Borning

1979: Alan Borning, *ThingLab*

1993: Sannella, Maloney, Freeman-Benson, Alan Borning, *Multi-way versus One-way Constraints in User Interfaces: Experience with the DeltaBlue Algorithm*

1998: Greg J. Badros, Alan Borning, *The Cassowary Linear Arithmetic Constraint Solving Algorithm: Interface and Implementation*

- 1997: Cassowary in Scwm (Scheme Constraints Window Manager)
- 2011: Cassowary in MacOS (Lion)
- 2014: Cassowary in Grid Style Sheets

delta blue

```

#lang typed/racket

(require "delta-blue-lisp.rkt")

(define x (var))
(define x-1)
(define z 2)
(define (show-variables)
  (print "x~s = ~s~s = ~s~s" x y z))

(define x-var (create-variable))
(define (add-const d)
  (define z-var (create-variable))
  (define c (create-constraint
             (list x-var y-var z-var)
             medium-strength
             #f
             (list
              #:x + y = z
              (create-method (lambda () (set! x (- z y))) 0)
              (create-method (lambda () (set! y (+ z y))) 1))
              (create-method (lambda () (set! z (+ x y))) 2)))
  (show-variables)
  (add-const d)
  (show-variables))

(define (show-variables)
  (print "add medium x + y = ~s~n")
  (add-const d)
  (show-variables))

(define d (create-constraint
            (list x-var)
            weak-strength
            #f
            (list
              #:x = 10
              (create-method (lambda () (set! x 10)) 0))))
  (show-variables))

(prin "add weak x = 10~n")
(add-const d)
(show-variables)

(define e (create-constraint
            (list y-var)
            strong-strength
            #f
            (list
              #:y = 20
              (create-method (lambda () (set! y 20)) 0)))
  (show-variables))

(prin "add strong y = 20~n")
(add-const e)
(show-variables)

(define f (create-constraint
            (list x-var y-var)
            strong-strength
            #f
            (list
              #:x + y = 10
              (create-method (lambda () (set! x (- 10 y))) 0)
              (create-method (lambda () (set! y (- 10 x))) 1)))
  (show-variables))

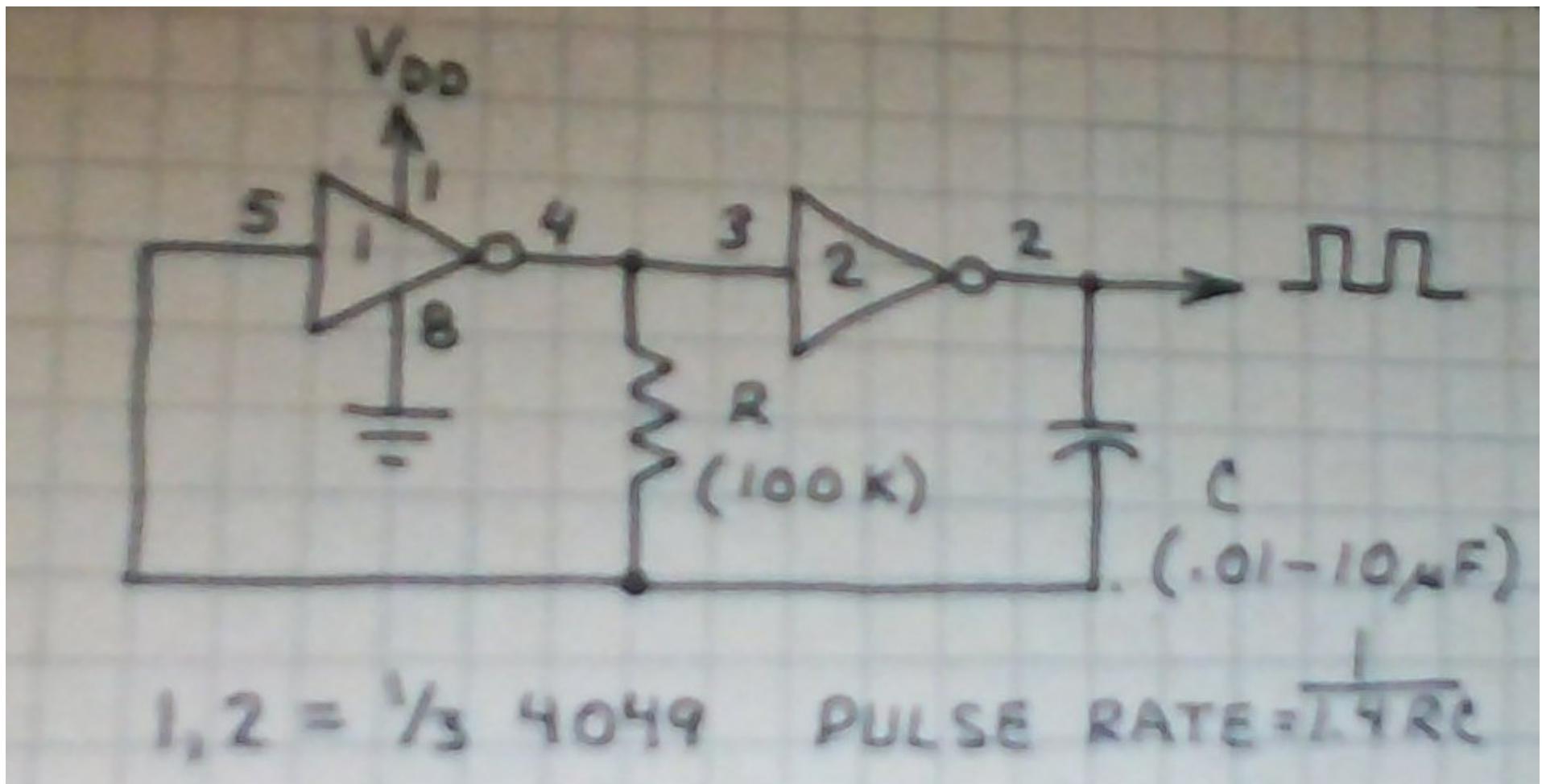
(prin "add strong x + y = 20~n")
(add-const f)
(show-variables)

(prin "remove strong constraint y=20~n")
(remove-const e)
(show-variables)

```

This statement is false.

A proposition? A paradox.

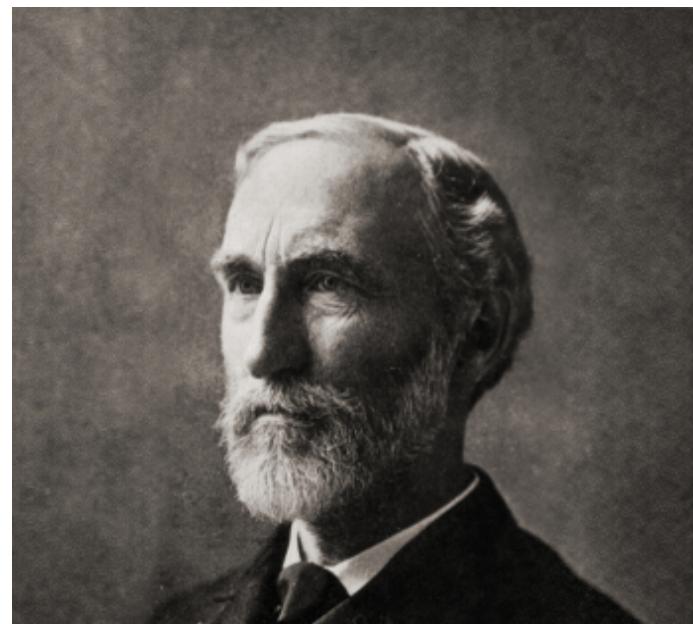


An oscillator.

Josiah Willard Gibbs (1839-1903)

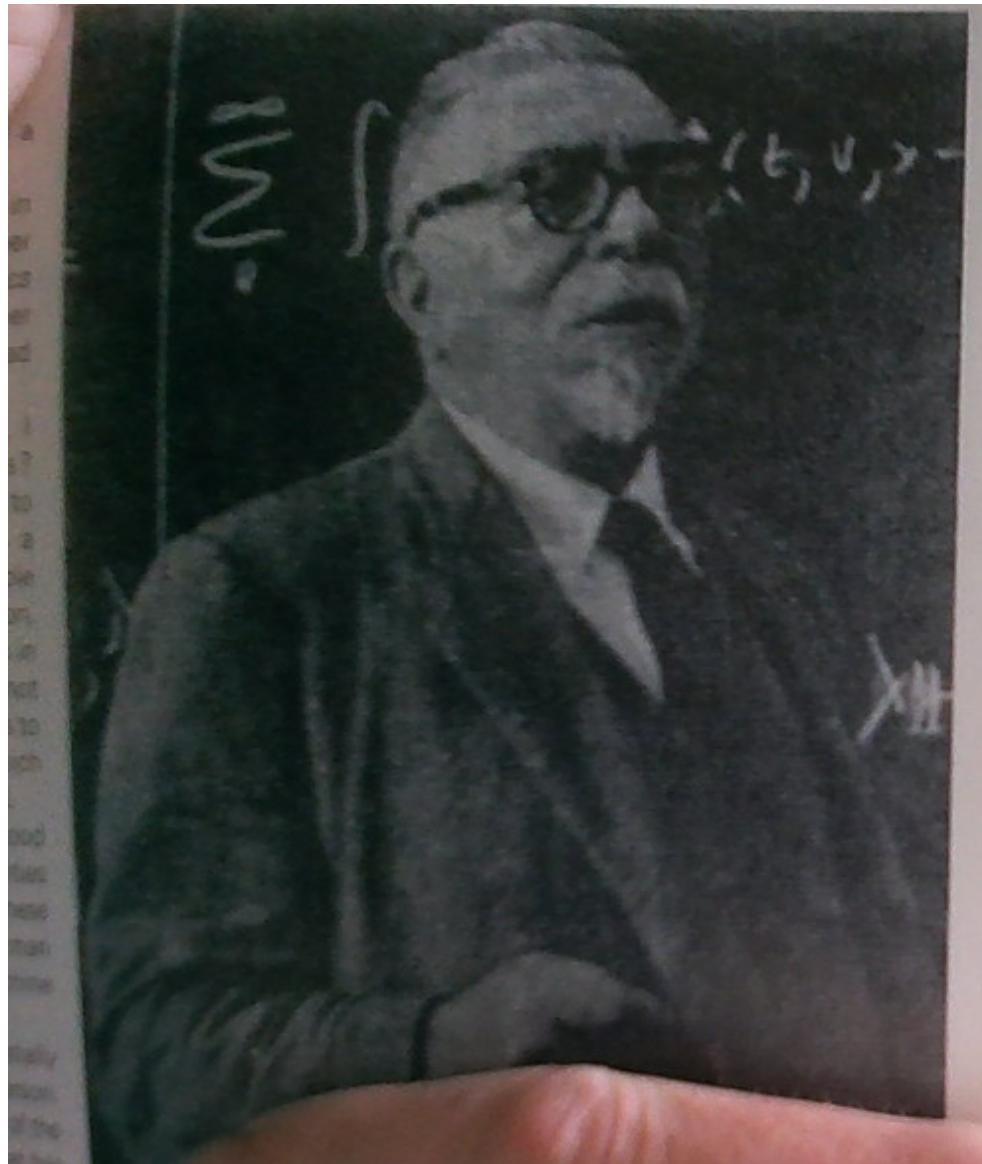
“Gibb’s innovation was to consider not one world, but all the worlds which are possible answers to a limited set of questions... His central notion concerned the extent to which answers that we may give to questions about one set of worlds are probable among a larger set of worlds.”, Norbert Wiener

“The measure of this probability is called entropy”



Norbert Wiener (1894-1964) — Cybernetics

1968ish: Philip Peterson, "Digital Norbert Wiener"



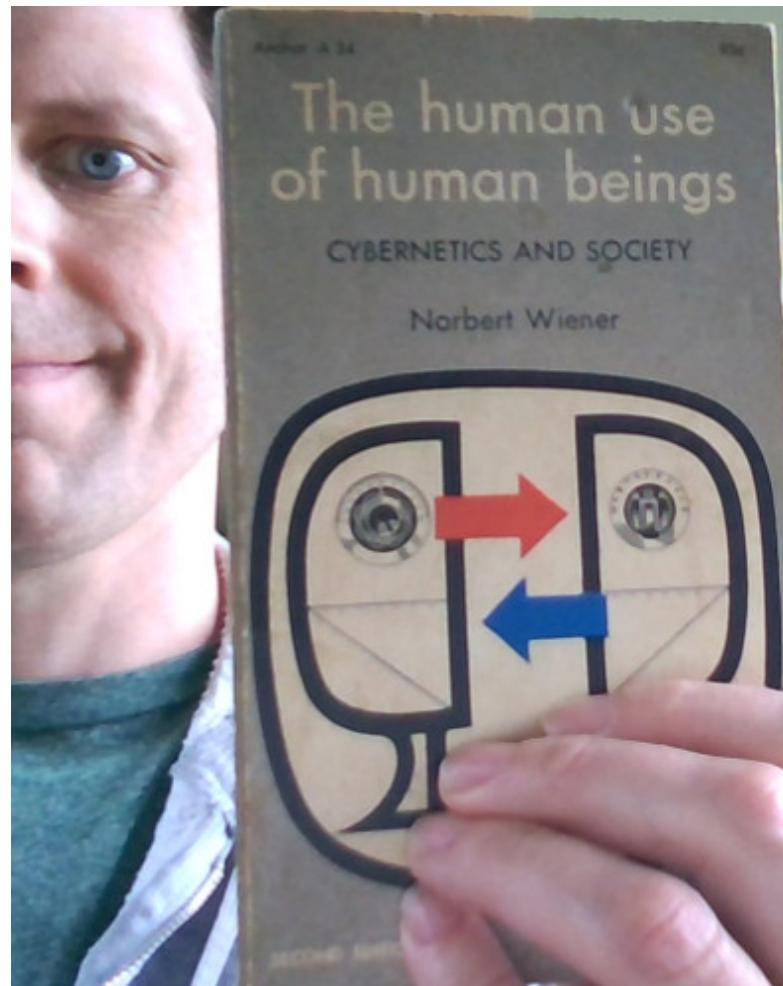
Claude Shannon (1916-2001) — Information Theory



The bit

1948: Norbert Wiener, "Cybernetics—Control and Communication in the Animal and Machine"

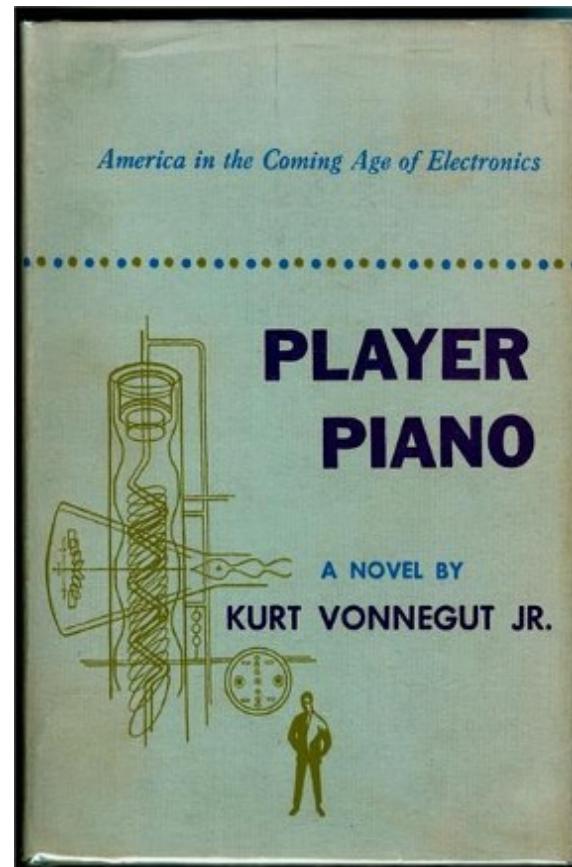
1950: Norbert Wiener, "The Human Use of Human Beings—Cybernetics and Society"



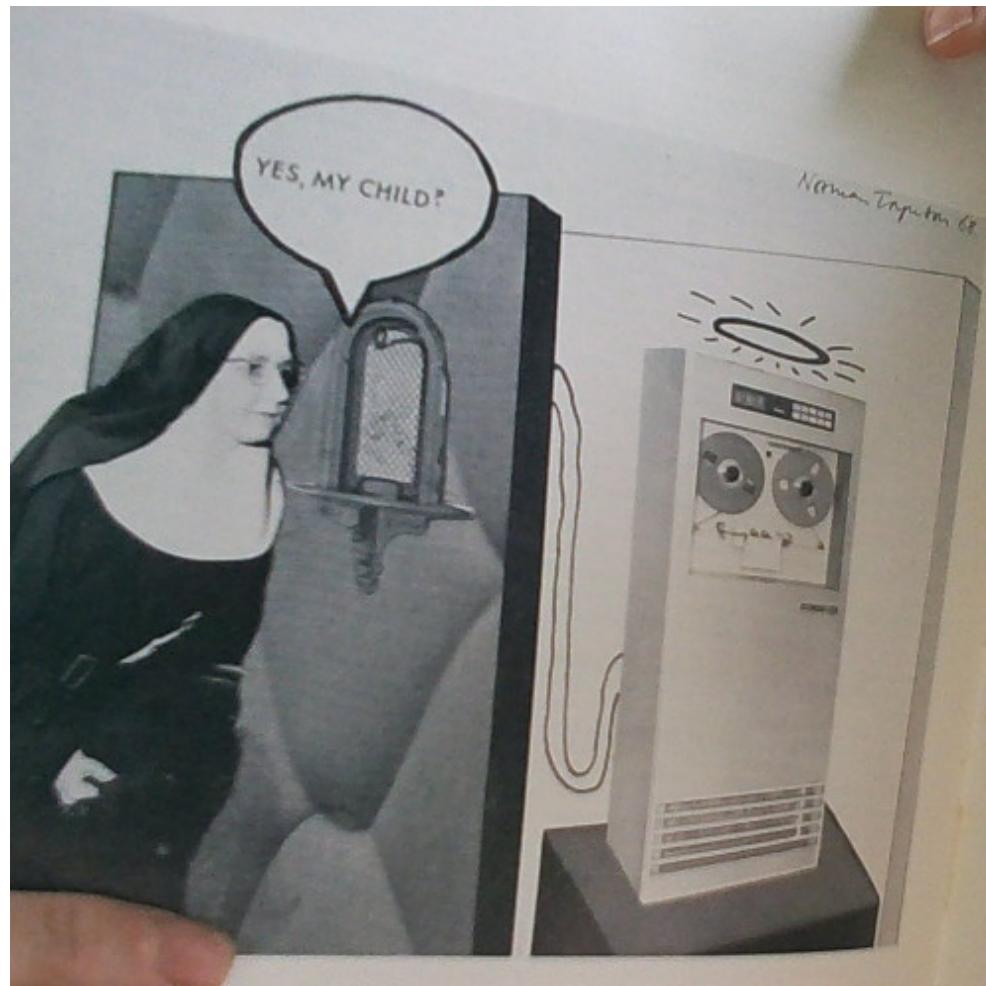
Cybernetics

Wiener, “It is my thesis that the physical functioning of the living individual and the operation of some of the newer communication machines are precisely parallel in the analogous attempts to control entropy through feedback.”

1952: Kurt Vonnegut, "Player Piano"



1968: Norman Toynton, "Yes, My Child"



1974: Brian Eno, "Seven Deadly Finns"



2013: Chris-Rachael Oesland, "White Chocolate Cyberman Heads"



Ludwig Wittgenstein (1889-1951), 1921

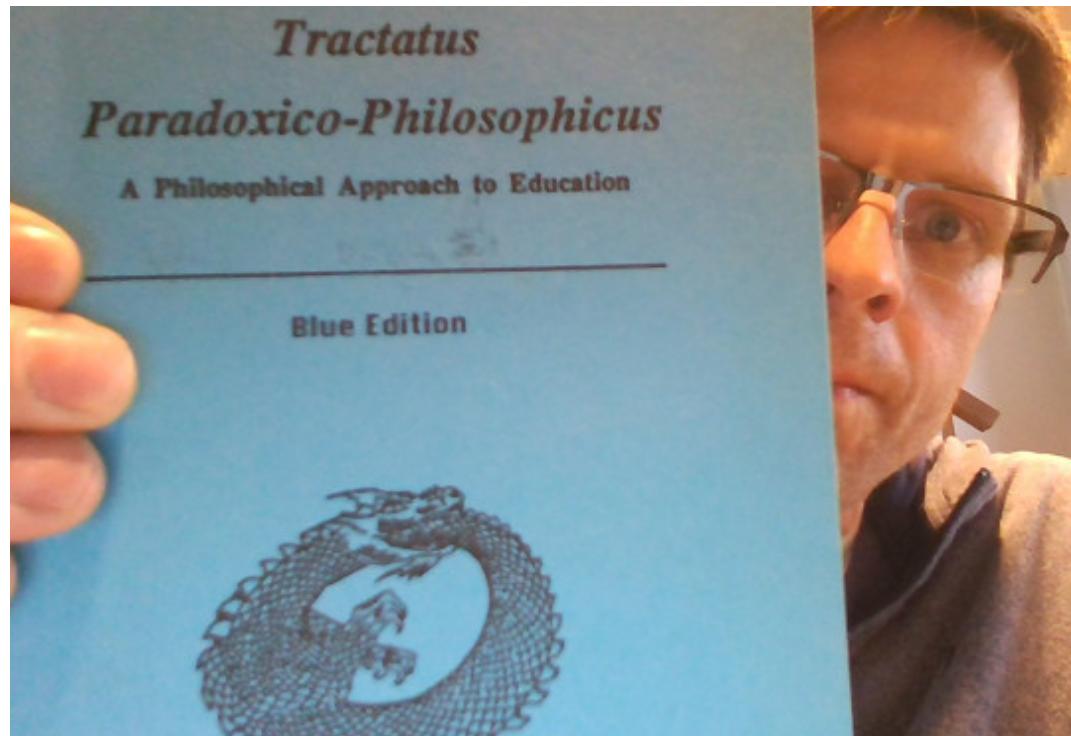
Tractatus
Logico-Philosophicus

By
LUDWIG WITTGENSTEIN

With an Introduction by
BERTRAND RUSSELL, F.R.S.

Ricardo Uribe

"intending to stimulate in the audience thoughts similar to mine."



1991 Yellow, 1994 Blue, Red Edition:
<http://bcl.ece.illinois.edu/Uribe/index.htm>



American Society for Cybernetics

The Wiener Gold Medal of the American Society for Cybernetics
is awarded to

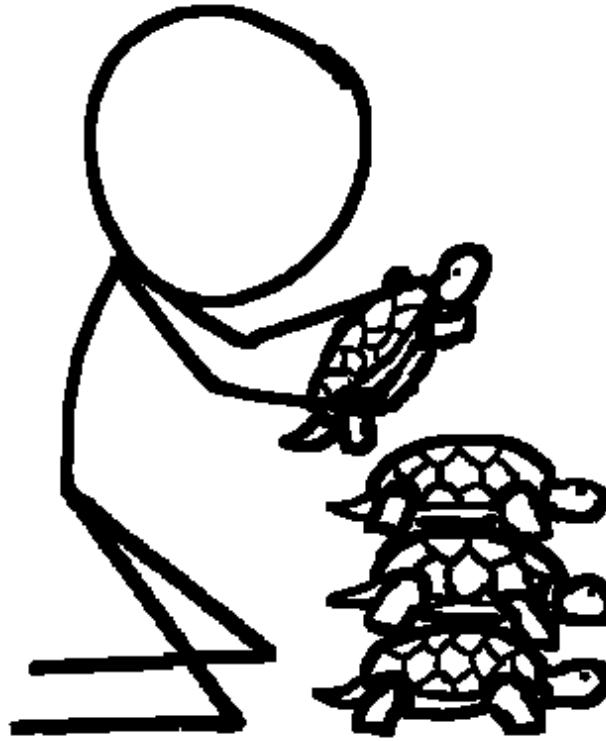
Ricardo Uribe

for outstanding and profound lifelong contributions to the nurturing of cybernetics.

Let's Get Cozy with Relations

by Anthony Carrico <acarrico@memebeam.org>

<https://github.com/acarrico/presentations/>



@Anthony_Carrico #vtfun @racketlang

Vermont Functional Users Group